

## CPSC 1155 – Lab 8

### Loops

#### Lab Introduction

This lab helps you practice with different forms of loops in pseudocode and C++.

#### Learning Objectives

At the end of this lab, you should be able to:

- trace a pseudocode/C++ program with loops
- write repetition statements in pseudocode and C++

#### Lab Readings

1. Chapter 5 – Loops
2. Pseudocode Lecture

#### Lab Instructions

Write your answers to the **Practice Questions** in a text editor (answer.txt).

For each **Problem Statement**, follow the steps below:

1. Read the problem statement and clarify the problem.
  - a. Break the problem into smaller problems if needed.
2. Determine the IPO.
  - a. Determine input, output, intermediate variables, constants, conditions, and **repetitions**.
  - b. Declare the variables and constants (data type + meaningful names).
  - c. Work out the problem by hand using typical input values. Determine the range of valid input values.
  - d. Determine the process.
3. Write a pseudocode as required.
4. Write a C++ program (use the given filename) that implements the pseudocode.
  - a. Add comments where needed. Make sure to use a comments header to reflect the intention of your program and name of the author (you) and the date the program was written.
  - b. Test, debug, and execute the program using typical values.

Submit according to the instruction in the "Lab Submission" section.

#### Learning Objectives

The goal of this lab is to practice with different forms of loops in C++.

## Practice Questions

1. [6] **Complete the Code.** Complete the given program that reads an integer  $n$  and prints the following diamond pattern with  $2 * n - 1$  asterisks at its widest point. This is the pattern when  $n$  is 3:

```
--*--  
_***_  
*****  
_***_  
--*--
```

```
#include <iostream>  
#include <cstdlib>  
using namespace std;  
int main()  
{  
    int n;  
    cin >> n;  
    // Draw top triangle  
    for (int left_stars = 0; left_stars < n; left_stars++)  
    {  
        for (int column = 0; column < 2 * n - 1 ; column++)  
        {  
            int first_star = n - 1 - left_stars ;  
            int last_star = n - 1 + left_stars;  
            if (column < first_star || column > last_star)  
            {  
                cout << "-";  
            }  
            else  
            {  
                cout << "*";  
            }  
        }  
        cout << endl;  
    }  
    // Draw bottom triangle  
    . . .  
    return 0;  
}
```

2. [6] **Trace a Program.** For each of the following, write a trace table to determine the execution of the loop. When choosing inputs in question a, make sure it loops at least 3 times. Explain the purpose of the loops in one or two sentences.

a. 

```
int value, product = 1;  
bool done = false;  
while (!done){  
    cin >> value;  
    if (value > 0)  
        product = product * value;  
    else  
        done = true;  
}  
cout << product;
```

```

b. for (int n = 2; n <= 3; n++){
    int p = 1;
    for (int i = 1; i <= 3; i++){
        p = p * n;
        cout << setw(6) << p;
    }
    cout << endl;
}
cout << endl;

```

### Problem Statements

3. [5] (trigo\_functions.cpp) You are required to calculate the `sin` value and `cos` value of degrees from 0 to 360 with increments of 10 degrees.

Write a loop that counts from 0 to 360 with increments of 10 (degree), and calculates the `sin` and `cos` for each degree. Round the values to keep 4 digits after the decimal point and display them as shown in the following table (left-aligned).

Degree	Sin	Cos
0	0.0000	1.0000
10	0.1736	0.9848
...		
350	-0.1736	0.9848
360	0.0000	1.0000

4. [6] (phone\_numbers.cpp) The following algorithm describes how to turn a string containing a ten-digit phone number (such as "4155551212") into a more readable string with parentheses and dashes, like this: "(415) 555-1212".

- Take the substring consisting of the first three characters and surround it with "(" and ")". This is the area code.
- Concatenate the area code, the substring consisting of the next three characters, a hyphen, and the substring consisting of the last four characters. This is the formatted number.

Translate this algorithm into a C++ program that reads a telephone number into a string variable, computes the formatted number, and prints it. Assume user input is valid.

5. [5] (sum\_of\_numbers.cpp) You are required to calculate the sum of integers from 1 to an input integer. The program should be able to repeat this process for any number of inputs. Assume user inputs are valid. You need to write **nested loops**.

- Ask the user to enter an integer (inputValue) and write a loop that counts from 1 to inputValue and adds the counter to sum on each iteration of the loop.
- Ask the user to enter the number of times the user wants to repeat this process (numberOfRepetitions). Write the code from part 'a' inside another loop that counts from 1 to the numberOfRepetitions.

Here is a sample run for 3 repetitions:

```

Enter the times you want to repeat: 3

Enter a number: 5
The sum of 1 to 5 is 15

Enter a number: 11
The sum of 1 to 11 is 66

```

```
Enter a number: 56
The sum of 1 to 56 is 1596
```

### Lab Submissions

Submit a zip folder named as yourName\_Lab8.zip to Brightspace. This folder should consist of a text file named **answer.txt** with your answers to Practice Questions and three **C++ programs** for Problem Statements.

Please make sure that all your C++ programs compile and run properly before submission.

### Marking Scheme

The marks are given in square brackets [ ] for each question.